

We claim:

Sub A² 1.

1. An apparatus for storing a computer-readable module for use in a host computer, where the module comprises:

5 a functional component for providing a predetermined programming functionality, and
a preprocessing interface component for providing introspective information about any requirements of the functional component in response to a request from the host computer.

10 2. The apparatus of claim 1 wherein the preprocessing interface comprises a command sequence that identifies the functional component.

15 3. The apparatus of claim 1 wherein the preprocessing interface comprises executable object code that identifies the functional component.

4. The apparatus of claim 1 wherein the request from the host computer occurs as part of an installation of the module in the host computer.

20 5. The apparatus of claim 1 wherein the request from the host computer occurs when the module is required by a second module stored at the host computer.

6. The apparatus of claim 1 wherein the introspective information provided by the preprocessing interface component comprises version information for the module.

Sub A² 7

7. The apparatus of claim 1 wherein the introspective information provided by the preprocessing interface component comprises identification of an external module required by the functional component.

5

8. The apparatus of claim 1 wherein the introspective information provided by the preprocessing interface component comprises identification of the type of functional component.

9. The apparatus of claim 1 wherein the introspective information provided by the preprocessing interface component comprises identification of at least a first resource at the host computer that will be modified by the functional component.

10. A computer system comprising a first computer, where the first computer comprises:

15

a plurality of plugin modules, including at least a first plugin module comprised of an introspection interface portion, an installation program component and a program behavior portion, and

20

a harness for evaluating the introspection interface portion of the first plugin module to determine if any resources are required for use of the program behavior portion of the first plugin module.

11. The computer system of claim 10 further comprising a second computer connected to the first computer, where the second computer includes a first resource

Sub A² >

required by the first plugin module that is transferred to the first computer upon request by the plugin installation program component.

12. The computer system of claim 10 further comprising a second computer
5 connected to the first computer, where the second computer includes a first resource required by the first plugin module that is transferred to the first computer upon request by the harness.

13. The computer system of claim 11, wherein the first resource comprises a
10 second plugin module comprised of an introspection interface portion and a program behavior portion, and wherein the harness is further structured to evaluating the introspection interface portion of the second plugin module to determine if any resources are required for use of the program behavior portion of the second plugin module.

14. The computer system of claim 10 wherein the harness comprises:
15 a loader component for loading a plugin module comprised of an introspection interface portion and a program behavior portion,
a validator component for interfacing with the introspection interface portion to identify any resources required by the plugin module, and
20 a finder component for surveying the computer system for any resources identified by the validator component.

Sub A² >

15. The computer system of claim 10 wherein the introspection interface portion is an executable script, command series or object-code module.

5 16. The computer system of claim 10 wherein the installation program component is an executable script, command series or object-code module.

17. The computer system of claim 10 wherein the computer system is a client/server system and the first plugin module is a client plugin module.

10 18. The computer system of claim 10 wherein the computer system is a client/server system and the first plugin module is a server plugin module.

19. A method of preprocessing a software module comprised of an interface portion and a program portion, comprising:

15 receiving a first software module,

querying the interface portion of the first software module to identify any resources required by the program portion, and

installing the first software module if all resources required by the program portion are available.

20

20. The method of claim 19, further comprising retrieving any resources required by the program portion that were identified as a result of querying the interface portion prior to invocation of the first software module.

